ECAP268 Computer System Architecture

1. Number system ?

ANS:- The number system is the numerical notation in which we use the digits or some other symbols for

representation. It gives a distinctive representation of every number. It provides us with the privilege to do simple operations of arithmetic like addition, subtraction, and division.

(i) Decimal Number System: When digits form a string, we call it a number. These decimal numbers have a base of D or 10 or Dec or radix. In a base-10 number ,we have the values which move from zero to nine: 0, 1, 2, 3, 4, 5,6, 7, 8, and 9 values.

Example :- A number like "4321" has four places, and each place can have the digits 0-9.

Example:- 123410 = 1x 103 + 2x 102 + 3x101 +4x100 = 1000 + 200 + 30 + 4 = 123410

Example 9876D = 9x 103 + 8x 102 + 7x101 +6x100 = 9000 + 800 + 70 + 6 = 9876D

(ii) Binary Number Systems:-_The term binary numbering formats corresponds to the system implemented in digital

computers to represent numbers. The base in binary number systems is 2 or ‘b’ or ‘B’ or ‘Bin ’.This uses two symbols: 0 and 1.Each bit in the number is weighted by the power of 2.

Example :- 01010101: The number of bits in this pattern is 8.

(iii) Octal Number System:-Computer mechanics often need to write out binary quantities, but in practice writing out a binary number such as 1001001101010001 is tedious and prone to errors. Therefore, binary quantities are written in a base-8 which is known as the octal representation. The base of octal numbers is 8 or ‘o’ .

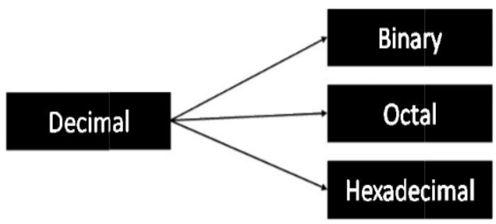
Example :- 123, 567, 7654

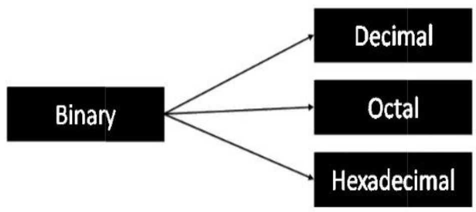
or ‘Oct’. In this 8 symbols are used: 0, 1, 2, 3, 4, 5, 6, 7. Each number is weighted by the power of 8.

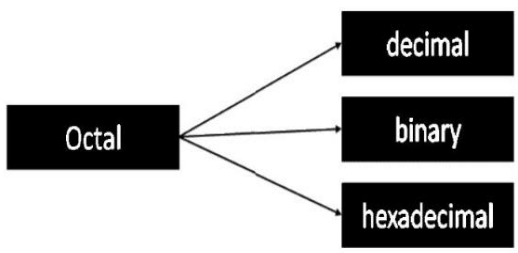
(iv).Hexadecimal Number System:_The hexadecimal number system base is 16 or ‘H’ or ‘Hex’.16 symbols are used: 10 digits and 6 letters { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10=A, 11=B, 12=C, 13=D, 14=E, 15= F}.Each symbol is weighted by the power of 16.

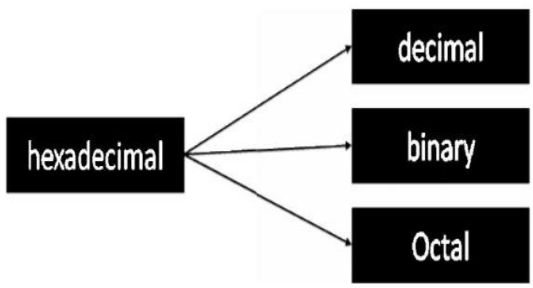
Example :- AB12, 876F, FFFF

2. Number System Conversion ?









3.Postulates of Algebraic Structures ?

ANS:-

(i) Closure:- A set is closed with respect to a binary operator if, for every pair of elements of S, the binary operator specifies a rule for obtaining a unique element of S. For example: The set of natural numbers N= {1,2,3,4........} is closed with respect to the binary operator + by the rules of arithmetic addition, since for any a, b∈N we obtain a unique c∈N by the operation of a+b=c. But the set of natural numbers is not closed with respect to binary operator minus (-) by the rules of arithmetic subtraction because 2-3=-1 and 2,3∈N while (-1) ∉ N.

(ii) Associative Law:- This law allows the removal of brackets from an expression and regrouping of the variables.

A + (B + C) = (A + B) + C = A + B + C (OR Associate Law)
A(B.C) = (A.B) C = A .B. C (AND Associate Law)

(iii) Commutative Law:- The order of application of two separate terms is not important

A . B = B . A The order in which two variables are AND’ ed makes no difference

A + B = B + A The order in which two variables are OR’ ed makes no difference

(iv) Identity element:- A term OR´ ed with a “0” or AND´ ed with a “1” will always equal that term

A + 0 = A A variable OR’ ed with 0 is always equal to the variable

A . 1 = A A variable AND’ ed with 1 is always equal to the variable

(v)Distributive Law:-This law permits the multiplying or factoring out of an expression.

A(B + C) = A.B + A.C (OR Distributive Law)

A + (B.C) = (A + B).(A + C) (AND Distributive Law)

(vi)Double Inversion Law:- A term that is inverted twice is equal to the original term

- (A’)’ = A A double complement of a variable is always equal to the variable

(vii) Annulment law:- A term AND´ed with a “0” equals 0 or OR´ed with a “1” will equal 1

- A . 0 = 0 A variable AND’ed with 0 is always equal to 0
- A + 1 = 1 A variable OR’ed with 1 is always equal to 1

(viii) Idempotent law :- An input that is AND´ed or OR´ed with itself is equal to that input

- A + A = A A variable OR’ed with itself is always equal to the variable
- A . A = A A variable AND’ed with itself is always equal to the variable

(ix)Complement law:- A term AND´ed with its complement equals “0” and a term OR´ed with its complement equals “1”

- A . A = 0 A variable AND’ed with its complement is always equal to 0
- A + A = 1 A variable OR’ed with its complement is always equal to 1

(x) Absorptive law:- This law enables a reduction in a complicated expression to a simpler one by absorbing like terms.

- A + (A.B) = A (OR Absorption Law)
- A(A + B) = A (AND Absorption Law)

4. De-Morgan's Theorem?

ANS:- De-Morgan's First Theorem: De-Morgan's First Theorem: When two (or more) input variables are AND' ed and negated, they are equivalent to the OR of the complements of the individual variables. Thus the equivalent of the NAND function and is a negative-OR function proving that A.B = A+B. Complementing the result of AND' ing variables together is equivalent to OR' ing the complements of the individual variables.

De-Morgan's Second Theorem: - DeMorgan's Second Theorem: It proves that when two (or more) input variables are OR'ed and negated, they are equivalent to the AND of the complements of the individual variables. Thus the equivalent of the NOR function and is a negative-AND function proving that (A+B)' = A'.B' and again we can show this using the following truth table.

5. K-Map ?

ANS:- It is a systematic method for simplifying the Boolean expressions. It produces simplest POS or SOP which is known as minimum expression .It is similar to TT because it represents all the possible values of IP variables and the resulting OP for each value. K-Map is an array of cells in which each cell represents a binary value of IP variables .It can be used for expressions with two, three, four and five variables.

2-variable k-map • For two variables, there are four min-terms.

3-variable k-map • For three variables, there are eight min-terms.

4-variable k-map • For four variables, there are sixteen min-terms.

K-Map SOP Minimization :-A minimized SOP contains the fewest possible terms with the fewest possible variables per term. After SOP expression is mapped, there are three steps in obtaining a minimum SOP expression.

1) Grouping of 1's.

2) Determine the product term of each group.

3) Summing the resulting product term.

K-Map POS Minimization:- After POS expression is mapped, there are three steps in obtaining a minimum POS expression.

1) Grouping of 0's.

2) Determine the sum term of each group.

3) Multiplying the resulting sum terms.

For a standard POS expression, 0 is placed on k-map for each sum term in the expression.
(A+B+C)(A+B'+C)(A'+B'+C)(A'+B+C)

6.Logic gates(and, or, not, nor , nand , xor, xnor)?

ANS:- Logic gates are the building blocks of any digital computer. Logic gates are electronic circuit having one or more than one input and only one output. The relationship between the input and the output is based on certain logic. The input any gate can have either 0 or 1. Based open the concept of logic gates, we get the output also 0 or 1.

(i)NOT gate :- The logic symbol for NOT gate is The expression for NOT gate is Output = (A)'. Here A is designated as the input provided.

(ii)OR gate:- The logic symbol for OR gate is The expression for OR gate is OUTPUT = (A+B). Here A and B are inputs provided.

(iv)AND gate :-The logic symbol for AND gate is The expression for AND gate is OUTPUT = (A * B). Here A and B are inputs provided.

(v) NOR gate:- The logic symbol for NOR gate is The expression for NOR gate is OUTPUT = (A + B)'

(v) NAND gate:- The expression for NAND gate is OUTPUT = (A * B)’. Here A and B are the inputs provided

(vi) XOR gate:- The logic symbol for XOR gate is The expression for XOR gate is OUTPUT = A * B’ + A’ * B. Here A and B are the inputs provided.

(vii)XNOR gate:- The expression for XNOR gate is OUTPUT = A * B + A’ * B’. Here A and B are the inputs provided to the gate.

7.Adder ,subtractor ,encoder ,decoder ,mux , demux ?

ANS:- **Half adder** :- The half adder needs two binary inputs and two binary outputs. The input variables designate the augend and addend bits (x and y). The output variables produce the sum and carry (S and C).

Full Adder:- A full adder is the combinational circuit that forms the arithmetic sum of three bits. It consists of three inputs and two outputs. Inputs (Two of the input variables denoted by x and y), represent the two significant bits to be added and third input, z represents the carry from the previous lower significant positions) and the outputs (S and C; the binary variable S gives the value of the least significant bit of the sum and the binary variable C given the output carry).

The Boolean expressions for full adder are $S=x'y'z+x'yz'+xy'z'+xyz$

$$C=xy+xz+yz$$

Subtractor: A subtractor is a digital circuit that performs subtraction of binary numbers. It is used to find the difference between two binary numbers.

Encoder: An encoder is a digital circuit that converts one or more inputs into a coded output. It is used to compress data or represent information in a more efficient way.

Decoder: A decoder is a digital circuit that converts a coded input into one or more output signals. It is used to decode information and activate specific outputs based on the input code.

Multiplexer (MUX): A multiplexer is a digital circuit that selects one of many input lines and forwards the selected input to a single output line. It is used to route data from multiple sources to a single destination.

Demultiplexer (DEMUX): A demultiplexer is a digital circuit that takes a single input and selects one of many output lines to forward the input to. It is used to distribute data from a single source to multiple destinations.

8. Flip-flop, sequential circuits ?

ANS:- **Flip-flop: -** A flip-flop is a digital circuit element that stores a binary value and is used as a basic memory unit in sequential logic circuits. It can be either a simple SR flip-flop, a D flip-flop, a JK flip-flop, or a T flip-flop.

Sequential Circuits:- Sequential circuits are digital circuits that have memory elements (flip-flops) and combinational logic. The output of a sequential circuit depends not only on the current inputs but also on the previous inputs and the state of the flip-flops. They can be synchronous, where the state changes on a clock signal, or asynchronous, where the state changes with specific input conditions. Sequential circuits are used in applications where past information is important, such as counters, shift registers, and state machines.

9. Construction of common bus system (with multiplexer and with three state bus buffers)?

Ans:- A common bus system can be constructed using multiplexers or three-state bus buffers.

- **Construction with multiplexers:** A multiplexer is a circuit that selects one of its inputs and outputs it to its output. In a common bus system, the multiplexers are used to select the data from one of the registers and output it to the bus. The number of multiplexers required is equal to the number of registers in the system.
- **Construction with three-state bus buffers:** A three-state bus buffer is a circuit that can be used to either pass data through or totristate it (disable it). In a common bus system, the three-state bus buffers are used to connect the registers to the bus. The three-state bus buffers are enabled when the data from a register is to be output to the bus, and they are disabled when the data from a register is not to be output to the bus.

10. Micro-operations and types ?

ANS:- Micro-operation is a simple operation performed on data stored in one or more registers. They are the basic building blocks of machine instructions.

Types of micro-operations include:-

Register transfer micro-operations: These micro-operations transfer data between registers. For example, a load micro-operation loads data from memory into a register, and a store micro-operation stores data from a register into memory.

Arithmetic micro-operations: These micro-operations perform arithmetic operations on data stored in registers. For example, an addition micro-operation adds two numbers stored in registers, and a subtraction micro-operation subtracts two numbers stored in registers.

Logic micro-operations: These micro-operations perform logical operations on data stored in registers. For example, an AND micro-operation performs the logical AND operation on two numbers stored in registers, and an OR micro-operation performs the logical OR operation on two numbers stored in registers.

Shift micro-operations: These micro-operations shift data stored in registers. For example, a left shift micro-operation shifts the bits in a register one position to the left, and a right shift micro-operation shifts the bits in a register one position to the right.

11. Computer registers ?

ANS:- Computer registers are small, fast storage locations within the central processing unit (CPU) of a computer. They are used to hold data temporarily during processing and play a crucial role in the execution of instructions.

12.Instruction cycle and instruction code ?

ANS:- Instruction Cycle:- The instruction cycle, also known as the fetch-execute cycle, is the basic process through which a computer's central processing unit (CPU) executes instructions. It consists of two main steps:

1.Fetch: The CPU fetches the next instruction from memory, using the program counter (PC) to determine the memory address of the instruction.

2.Execute: The fetched instruction is then decoded and executed by the CPU, which performs the specific operations specified by the instruction.

Instruction Code: The instruction code, also referred to as the machine code or opcode, is a binary representation of an instruction in a computer program. It is a set of binary digits that directly corresponds to a specific operation to be performed by the CPU.

13.BSA instruction and risc instruction?

ANS:- BSA is a machine-level instruction used in certain computer architectures to perform subroutine calls. It transfers control to a subroutine while saving the return address of the calling program.

RISC Instruction (Reduced Instruction Set Computing): RISC is a computer architecture design that emphasizes the use of a small and simple set of instructions. RISC processors have instructions that are designed to perform simple operations efficiently.

14.Stack and it's operations ?

ANS:- Stack is a linear data structure that follows the Last-In-First-Out (LIFO) principle. It works like a collection of elements with two main operations:

1. Push: Adds an element to the top of the stack.
2. Pop: Removes the top element from the stack.

15.Program counter, accumulator ,stack pointer ?

ANS:- Program Counter: The program counter (PC) is a register in a computer's central processing unit (CPU) that holds the memory address of the next instruction to be fetched and executed. It keeps track of the sequence of instructions in a program, ensuring their proper execution in the correct order.

Accumulator: The accumulator is a special register in the CPU used for arithmetic and logic operations. It temporarily stores intermediate results during calculations and serves as a primary working register for these operations.

Stack Pointer: The stack pointer is a register in the CPU that keeps track of the top of the stack in memory. It is used for managing the program's call stack during function calls and returns, ensuring proper execution flow and memory management.

16. parallel processing, pipelining ?

Ans:- Parallel Processing: Parallel processing is a computing technique where multiple processors or cores work together simultaneously to execute tasks or parts of a task concurrently. It allows for faster and more efficient data processing by dividing a complex task into smaller sub-tasks and processing them in parallel.

Pipelining: Pipelining is a technique used to improve the throughput and performance of a computer's central processing unit (CPU). It breaks down the instruction execution process into smaller stages, and different instructions are processed in parallel stages of the pipeline.

17. Main Memory, RAM and ROM chips, Auxiliary memory, Associative Memory ,Cache Memory, Virtual Memory.?

ANS:- Main Memory: Main memory, also known as primary memory or RAM (Random Access Memory), is a volatile memory used to temporarily store data and program instructions that the CPU can access quickly. It is the main workspace for running applications and processing data during the computer's operation.

RAM (Random Access Memory): RAM is a type of main memory that allows the CPU to read and write data quickly. It provides fast access to data, but its contents are lost when the computer is powered off.

ROM (Read-Only Memory) chips: ROM is a type of non-volatile memory that stores permanent instructions or data that cannot be changed or modified by the user. It contains the firmware and initial instructions needed to start up the computer.

Auxiliary Memory: Auxiliary memory, also known as secondary memory or external memory, is non-volatile memory used to store data and programs for long-term storage. Examples include hard disk drives (HDDs) and solid-state drives (SSDs).

Associative Memory: Associative memory, also known as content-addressable memory (CAM), is a type of specialized memory used in high-speed search and retrieval operations. It allows data to be accessed based on its content rather than its memory address.

Cache Memory: Cache memory is a small, fast memory located between the CPU and main memory. It stores frequently accessed data and instructions to reduce access time and improve overall system performance.

Virtual Memory: Virtual memory is a memory management technique that allows the computer to use part of the hard disk as an extension of the main memory. It allows the computer to run larger programs and handle more data than the physical RAM alone can accommodate.

18.Peripheral Devices , Strobe and handshaking ?

ANS:- **Peripheral Devices:** Peripheral devices are external hardware components connected to a computer that expand its functionality and allow users to input data, output information, or perform additional tasks. Examples include keyboards, mice, monitors, printers, scanners, and external storage devices like USB drives.

Strobe: In digital systems, a strobe signal is a brief pulse or control signal that indicates the availability or readiness of data for processing. It synchronizes the timing of operations between different parts of a system and is often used to control the flow of data between devices.

Handshaking: Handshaking is a communication process between two devices to establish a connection or synchronize data transmission. It involves exchanging predefined signals or sequences of signals to confirm that both devices are ready to send or receive data. Handshaking ensures proper coordination and data integrity during communication between devices. It is commonly used in serial communication protocols.

19. Asynchronous and synchronous Serial Transfer ?

Ans:- Asynchronous Serial Transfer: Asynchronous serial transfer is a method of data communication where individual data bits are transmitted without a shared clock signal between sender and receiver.
Synchronous Serial Transfer: Synchronous serial transfer is a method of data communication where data is transmitted in a continuous stream with a shared clock signal between sender and receiver.
- The sender and receiver are synchronized with a common clock signal, ensuring that data bits are transmitted at precise intervals.

20. Verilog and VHDL ?

Ans:- Verilog: Verilog is a hardware description language (HDL) used for designing and modeling digital systems at the behavioral, structural, and gate levels. It is widely used in digital design and verification.

VHDL (VHSIC Hardware Description Language): VHDL is another hardware description language used for designing digital systems. It was developed as part of the VHSIC (Very High-Speed Integrated Circuit) program and is widely used in the hardware design and verification industries.

By- Topers Of LPU